98

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/618,500	07/11/2003	Christopher Y. Blaicher	149-0105US (02-014-US)	5384

29855 7590 12/26/2006

WONG, CABELLO, LUTSCH, RUTHERFORD & BRUCCULERI, L.L.P.
20333 SH 249
SUITE 600
HOUSTON, TX 77070

EXAMINER	
SAEED, USMAAN	
ART UNIT	PAPER NUMBER
2166	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	12/26/2006	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

DEC 2 6 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/618,500
Filing Date: July 11, 2003
Appellant(s): BLAICHER, CHRISTOPHER Y.

Coe F. Miles
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 10/13/2006 appealing from the Office action mailed on 5/15/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Applicant's Admitted Prior Art (AAPA).

5,247,665	Matsuda et al.	9-1993
5,274,805	Ferguson et al.	12-1993

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-12, 16-24, 27-37, and 40-52 are rejected under 35 U.S.C 103(a) as being unpatentable over **Applicants Admitted Prior Art** (**AAPA** hereinafter) in view of **Matsuda et al.** (**Matsuda** hereinafter) (U.S. Patent No. 5,247,665).

With respect to claim 1, **AAPA** teaches **a data sort method, comprising:**

"obtaining a plurality of data records and, for each data record" as an object typically includes one or more records (**AAPA** Paragraph 0002). Sort routine 100 reads and pads a record from the object being sorted (**AAPA** Paragraph 0003).

"extracting key information

expanding the extracted key information" as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003). Therefore these lines teach us about getting the keys and then expanding them.

"storing the expanded key information in a key record" as once padded, the record is written to an intermediate file (block 110) (**AAPA** Paragraph 0003). The time required to write and read an intermediate file having expanded sort keys can consume a significant portion of the total time needed to sort the object (**AAPA** Paragraph 0003). The examiner interprets the key record as an intermediate file, which stores expanded sort keys.

"sorting the plurality of key records based on the expanded key information" as a sort utility is invoked that reorders and then stores the padded records in a result file (**AAPA** Paragraph 0003).

"reorganizing the plurality of data records to correspond to the order of the sorted plurality of key records" as one or more fields are designated as a sort key and that sorting reorders an object's records based on the value of the records' sort keys (**AAPA** Paragraph 0002). The object's records are being sorted based on the records' sort keys.

"storing the reorganized plurality of data records without their associated expanded key information to a working storage" as each sorted and padded record is then retrieved from the result file, unpadded and reloaded into the object (blocks 125, 130 and 135) (**AAPA** Paragraph 0003). The records in the object are storing the unpadded keys instead of padded/expanded key information.

AAPA teaches the elements of claim 1 as noted above but does not explicitly teach the **"wherein the expanded key information is not stored in intermediate storage."**

However, **Matsuda** discloses, **"wherein the expanded key information is not stored in intermediate storage"** as (Matsuda Abstract & Col 6, Lines 52-67 & Fig 2). These lines and fig 2 discloses the extraction and expansion of the key fields by adding the addresses of main memory. These expanded keys are not being stored in any intermediate storage but are being stored in the main memory.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to reduce the load factors of the CPU and the bus system and the processing performance is greatly improved by not transferring the result from main memory to the magnetic disk/intermediate storage (**Matsuda** Col 7, Lines 36-40).

Claims 16, 27 and 40 are same as claim 1 except claims 27 and 40 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

Claims 22 and 24 contain the elements of claim 1 and are rejected for the same reasons as applied hereinabove.

With respect to claim 2, **AAPA** teaches, **"the method of claim 1, wherein the act of obtaining comprises obtaining data records from one or more storage devices"** as reading the object from external storage (**AAPA** paragraph 0002).

Claims 17, and 28 are same as claim 2 except claim 28 sets forth the claimed invention as a program storage device are rejected for the same reasons as applied hereinabove.

With respect to claim 9, **AAPA** teaches **"the method of claim 1, wherein the act of expanding comprises adjusting each key field to a fixed length"** as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003).

Claims 35, and 50 are essentially the same as claim 9 except they set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 10, **AAPA** teaches **"the method of claim 1, wherein the act of storing the expanded key information in a key record further comprises, associating a value with each key record that identifies the data record from which the expanded key information was extracted"** as one or more fields are designated as a sort key and that sorting reorders an object's records based on the value of the records' sort keys (**AAPA** Paragraph 0002). The value of the records sort key is identifying the object/data records.

Claims 23, 36, and 51 are same as claim 10 except claims 36 and 51 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 11, **AAPA** teaches **"the method of claim 10, wherein the act of storing the expanded key information in a key record does not comprise storing a data field from the data record associated with the key record"** as once padded, the record is written to an intermediate file (block 110) (**AAPA** Paragraph 0003). The time required to write and read an intermediate file having expanded sort keys can consume a significant portion of the total time needed to sort the object (**AAPA** Paragraph 0003). Therefore the intermediate file has expanded sort keys, which are needed to sort the object/data records.

Claims 37, and 52 are essentially the same as claim 11 except they set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 3, **AAPA** does not explicitly teach **"the method of claim 1, wherein the act of extracting comprises: determining a starting location for a first key field; and calculating the starting location of a subsequent key field based on the determined starting location of the first key field."**

However, **Matsuda** discloses **"the method of claim 1, wherein the act of extracting comprises: determining a starting location for a first key field"** as adds the start location data (on the main memory 9) of the record in which the extracted key is stored to the key field Ki to for input data DI, and outputs the data DI to the RAPU 15. At this time, the controller 13 adds a flag indicating that the data is a key field Ki or an identifier Ai to the output data in synchronism with data output to the RAPU 15 (**Matsuda** Col 9, Lines 59-65). These lines teach the starting location of the data in which extracted key is stored and the flag indicates that the data is a key field. **"calculating the starting location of a subsequent key field based on the determined starting location of the first key field"** as in the key extraction processing, if the processing target file cannot be stored in the I-BUF at once, and hence mode for fetching the file in the I-BUF in units of elements of the file is set, every time key field extraction processing is completed for the element held in the I-BUF, the next element to be processed is fetched in the I-BUF. Such key field extraction

processing is repeated until processing of the entire target file is completed (**Matsuda** Col 20, Lines 7-11). Each record length and each key field length are constant, the start location data (on the main memory 9) of each record can be obtained by multiplying the record length by the number of records (**Matsuda** Col 9, Lines 4-8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the starting location of the fields.

Claims 29, and 44 are essentially the same as claim 3 except they set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 4, **AAPA** does not explicitly teach **"the method of claim 3, wherein the act of determining comprises obtaining the starting location of the first key field from a sort control card."**

However, **Matsuda** discloses **"the method of claim 3, wherein the act of determining comprises obtaining the starting location of the first key field from a sort control card"** as the controller extracts a processing target key field Ki from each record of the processing target file stored in the first area of the main memory 9, and

adds the start location data (on the main memory 9) of a record having the key field to the key field Ki as an identifier Ai. In this embodiment, since each record comprises a plurality of key fields, and each record length and each key field length are constant, the start location data (on the main memory 9) of each record can be obtained by multiplying the record length by the number of records (**Matsuda** Col 8, Lines 66-68 & Col 9, Lines 1-8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the starting location of the fields.

Claims 18, 30, and 45 are same as claim 4 except claims 30 and 45 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 5, **AAPA** does not explicitly teach, **"the method of claim 4, wherein the sort control card comprises a parameter list."**

However, **Matsuda** discloses **"the method of claim 4, wherein the sort control card comprises a parameter list"** as the controller is constituted by, e.g., a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., and receives

parameters from the CPU 1. The parameters include: data representing the position of a result identifier string stored in the second area of the main memory 9 (addresses in the second area of the main memory 9); logical data concerning a processing target file stored in the first area of the main memory 9 (file data such as a file format, a block length, and a record length); logical data concerning an output file (a file format, a block length, a record length, and the like); and physical data representing the storage location of an output file (output file data such as a location on a disk and size) (**Matsuda** Col 10, Lines 52-62).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to rearranges the respective records in accordance with these parameters and thus forming an output file (**Matsuda** Col 10, Lines 63-68).

Claims 19, 31, and 46 are same as claim 5 except claims 31 and 46 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 6, **AAPA** does not explicitly teach, **"the method of claim 4, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record."**

However, **Matsuda** discloses **"the method of claim 4, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record"** as the logical data of each of the processing target file and the output file include a file format, a block length, a record length, the respective key field positions and lengths of a multi-key field (**Matsuda** Col 12, Lines 10-14). Adding an identifier as a number or relative position data of each record to the key field, performing an operation designated by the operation command using an obtained pair of the key field and the identifier as a processing unit (**Matsuda** Col 6, Lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the position of the key fields.

Claims 32, and 47 are essentially the same as claim 6 except they set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 7, **AAPA** does not explicitly teach, **"the method of claim 4, wherein the sort control card further indicates a data type for each key field in a record."**

However, **Matsuda** discloses **"the method of claim 4, wherein the sort control card further indicates a data type for each key field in a records"** as the controller is constituted by, e.g., a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., and receives parameters from the CPU 1. The parameters include: data representing the position of a result identifier string stored in the second area of the main memory 9 (addresses in the second area of the main memory 9); logical data concerning a processing target file stored in the first area of the main memory 9 (file data such as a file format, a block length, and a record length); logical data concerning an output file (a file format, a block length, a record length, and the like); and physical data representing the storage location of an output file (output file data such as a location on a disk and size) (**Matsuda** Col 10, Lines 52-62). Examiner interprets the file format as data type.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to rearranges the respective records in accordance with these parameters and thus forming an output file (**Matsuda** Col 10, Lines 63-68) and data type being one of the parameters.

Claims 20, 33, and 48 are same as claim 7 except claims 33 and 48 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 8, **AAPA** does not explicitly teach, **"the method of claim 7, wherein the sort control card further indicates a sort order for each key field in a record."**

However, **Matsuda** discloses **"the method of claim 7, wherein the sort control card further indicates a sort order for each key field in a record"** as upon reception of an operation command for sorting (ascending order/descending order) or a relational operation from an input mechanism on a terminal side (**Matsuda** Col 9, Lines 35-37).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66).

Claims 21, 34, and 49 are same as claim 8 except claims 34 and 49 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 12, **AAPA** does not explicitly teach, **"The method of claim 1, wherein the working storage comprises one or more direct access storage devices."**

However, **Matsuda** discloses **"The method of claim 1, wherein the working storage comprises one or more direct access storage devices"** as in FIG. 5, a bus

connection may be designed to allow the DBPU 27 to directly access a magnetic disk unit under the control of the CPU, so that a processing target file is stored in the local memory, and the start location data of a record having a designated key field in the local memory may be added to the corresponding key as an identifier (**Matsuda** Col 16, Lines 62-68). External storage means for storing files comprising records having a plurality of data fields each assignable as a key field (**Matsuda** Col 2, Lines 5-7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation, and which can efficiently execute sophisticated arithmetic operations even if a target file is dispersed in a plurality of magnetic disk units connected to different input and output channel devices (**Matsuda** Col 1, Lines 63-67 & Col 2, Lines 1-2).

Claims 41, and 42 are essentially the same as claim 12 except they set forth the claimed invention as a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 43, **AAPA** does not explicitly teach **"the sorting system of claim 40, wherein the processing means comprises two or more communicatively coupled computer processors."**

However, **Matsuda** discloses **"the sorting system of claim 40, wherein the processing means comprises two or more communicatively coupled computer processors"** as the controller 13 is constituted by a microprocessor in this embodiment. For example, a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., may be used as this microprocessor (**Matsuda** Col 8, Lines 58-61). The controller 21 is constituted by, e.g., a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., and receives parameters from the CPU 1 (**Matsuda** Col 10, 49-51).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66).

Claims 13-15, 25-26, 38-39, and 53-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Applicants Admitted Prior Art** in view of **Matsuda et al.** (U.S. Patent No. 5,247,665) as applied to claims 1-12, 16-24, 27-37, and 40-52 above, further in view of **Ferguson et al. (Ferguson** hereinafter) (U.S. Patent No. 5,274,805).

With respect to claim 13, **AAPA and Matsuda** do not explicitly disclose **"the method of claim 1, further comprising repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records."**

However, **Ferguson** discloses **"the method of claim 1, further comprising repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records"** as In terms of a tree structure, the substrings are formatted as "leaf nodes", in that they comprise keys and pointers to records. To complete the upper levels of the tree structure, back to a root node, the logically sorted substrings are read in order from the storage system into memory, and a table of branch node key records is built up by reading key records from the substrings at node-sized intervals. A pointer to each such key record is determined and stored in the branch node table with the search key from the key record. When the branch node table is full, it is written out to the storage system, and a new branch node table is begun. The process is continued until all substrings are read. The process is then repeated, except that the first level of branch nodes are read from the storage system into memory and a second level of branch nodes are constructed. The process continues until a single root node is constructed (**Ferguson** Col 5, Lines 24-40).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Ferguson's** teaching would have allowed **AAPA and Matsuda** to sort key records first, and then build a tree based on keys extracted at intervals from sorted key records (**Ferguson** Col 3, Lines 2-4).

Claims 25, 38, and 53 are same as claim 13 except claims 38 and 53 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 14, **AAPA and Matsuda** does not explicitly teach **"the method of claim 13, further comprising merging the two or more plurality of reorganized data records."**

However, **Ferguson** discloses **"the method of claim 13, further comprising merging the two or more plurality of reorganized data records"** as after the generation of all necessary strings, at least two strings at a time are read back into memory and then merged into sorted order (this example is of 2-way merging; it is known in the art to extend this concept to N-way merging). An example of this process is diagrammatically shown in FIG. 2. The merged string is then written out to the storage system. Such merging continues for subsequent passes until only a single, sorted string remains that contains all of the key records (**Ferguson** Col 3, Lines 18-27).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Ferguson's** teaching would have allowed **AAPA and Matsuda** to require fewer storage system access and hence is generally faster (**Ferguson** Col 5, Lines 42-45) by merging the reorganized data records.

Claims 26, 39, and 54 are same as claim 14 except claims 39 and 54 set forth the claimed invention as a program storage device and a system and are rejected for the same reasons as applied hereinabove.

With respect to claim 15, **AAPA and Matsuda** do not explicitly teach **"the method of claim 14, wherein the act of obtaining a plurality of data records comprises obtaining a plurality of DB2 data records and the act of merging further comprises reloading the merged plurality of reorganized data records into the DB2 data object."**

However, **Ferguson** discloses **"the method of claim 14, wherein the act of obtaining a plurality of data records comprises obtaining a plurality of DB2 data records and the act of merging further comprises reloading the merged plurality of reorganized data records into the DB2 data object"** as it should be noted that the sort was conducted entirely "in place" in that no working space was set aside on the storage system to temporarily store output data. All processed data is written back into the same storage system area from which the data was originally read. The inventive method therefore provides a way of sorting very large databases. This is very useful, for example, when sorting data on a storage system that has no excess storage space available (**Ferguson** Col 9, Lines 64-68 & Col 10, Lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because

Ferguson's teaching would have allowed **AAPA and Matsuda** to require fewer storage system access and hence is generally faster (**Ferguson** Col 5, Lines 42-45).

With respect to claim 55, **AAPA** teaches **"a data sort method, comprising:**

"obtaining a plurality of data records from a database object, for each of the plurality of data records" as an object typically includes one or more records (**AAPA** Paragraph 0002). Sort routine 100 reads and pads a record from the object being sorted (**AAPA** Paragraph 0003).

"extracting key information,

expanding the extracted key information" as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003). Therefore these lines teach us about getting the keys and then expanding them.

"storing the expanded key information in a key record" as once padded, the record is written to an intermediate file (block 110) (**AAPA** Paragraph 0003). The time required to write and read an intermediate file having expanded sort keys can consume a significant portion of the total time needed to sort the object (**AAPA** Paragraph 0003). The examiner interprets the key record as an intermediate file, which stores expanded sort keys.

"sorting the plurality of key records based on the expanded key information" as a sort utility is invoked that reorders and then stores the padded records in a result file (**AAPA** Paragraph 0003).

"reorganizing the plurality of data records to correspond to the order of the sorted plurality of key records" as one or more fields are designated as a sort key and that sorting reorders an object's records based on the value of the records' sort keys (**AAPA** Paragraph 0002). The object's records are being sorted based on the records' sort keys.

"storing the reorganized plurality of data records without their associated expanded key information in a working storage" as each sorted and padded record is then retrieved from the result file, unpadded and reloaded into the object (blocks 125, 130 and 135) (**AAPA** Paragraph 0003). The records in the object are storing the unpadded keys instead of padded/expanded key information.

AAPA discloses the elements of claim 55 as noted above but does not explicitly teach the steps of **"wherein the expanded key information is not stored in intermediate storage"**

"repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records"

"merging the at least two plurality of reorganized data records"

"re-loading the merged plurality of reorganized data records into the database object."

However, **Matsuda** discloses, **"wherein the expanded key information is not stored in intermediate storage"** as (Matsuda Abstract & Col 6, Lines 52-67 & Fig 2). These lines and fig 2 discloses the extraction and expansion of the key fields by adding

the addresses of main memory. These expanded keys are not being stored in any intermediate storage but are being stored in the main memory.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to reduce the load factors of the CPU and the bus system and the processing performance is greatly improved by not transferring the result from main memory to the magnetic disk/intermediate storage (**Matsuda** Col 7, Lines 36-40).

AAPA and Matsuda disclose the elements of claim 55 as noted above but does not explicitly teach the steps of **"repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records"**

"merging the at least two plurality of reorganized data records"

"re-loading the merged plurality of reorganized data records into the database object."

However, **Ferguson** discloses, **"repeating the acts of obtaining, sorting, reorganizing and storing for at least a second plurality of data records"** as In terms of a tree structure, the substrings are formatted as "leaf nodes", in that they comprise keys and pointers to records. To complete the upper levels of the tree structure, back to a root node, the logically sorted substrings are read in order from the storage system into memory, and a table of branch node key records is built up by reading key records from the substrings at node-sized intervals. A pointer to each such key record is determined and stored in the branch node table with the search key from

the key record. When the branch node table is full, it is written out to the storage system, and a new branch node table is begun. The process is continued until all substrings are read. The process is then repeated, except that the first level of branch nodes are read from the storage system into memory and a second level of branch nodes are constructed. The process continues until a single root node is constructed (**Ferguson** Col 5, Lines 24-40).

"merging the at least two plurality of reorganized data records" as after the generation of all necessary strings, at least two strings at a time are read back into memory and then merged into sorted order (this example is of 2-way merging; it is known in the art to extend this concept to N-way merging). An example of this process is diagrammatically shown in FIG. 2. The merged string is then written out to the storage system. Such merging continues for subsequent passes until only a single, sorted string remains that contains all of the key records (**Ferguson** Col 3, Lines 18-27).

"re-loading the merged plurality of reorganized data records into the database object" as it should be noted that the sort was conducted entirely "in place" in that no working space was set aside on the storage system to temporarily store output data. All processed data is written back into the same storage system area from which the data was originally read. The inventive method therefore provides a way of sorting very large databases. This is very useful, for example, when sorting data on a storage system that has no excess storage space available (**Ferguson** Col 9, Lines 64-68 & Col 10, Lines 1-4).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Ferguson's** teaching would have allowed **AAPA and Matsuda** to require fewer storage system access and hence is generally faster (**Ferguson** Col 5, Lines 42-45) by merging the reorganized data records and to sort key records first, and then build a tree based on keys extracted at intervals from sorted key records (**Ferguson** Col 3, Lines 2-4).

Claim 61 is same as claim 9 and is rejected for the same reasons as applied hereinabove.

Claim 62 is same as claim 10 and is rejected for the same reasons as applied hereinabove.

Claim 63 is same as claim 11 and is rejected for the same reasons as applied hereinabove.

With respect to claim 56, **AAPA** does not explicitly teach **"the data sort method of claim 55, wherein the act of extracting comprises obtaining the starting location of a first key field in a data record from a sort control card."**

However, **Matsuda** discloses **"the data sort method of claim 55, wherein the act of extracting comprises obtaining the starting location of a first key field in a data record from a sort control card"** as the controller extracts a processing target key field Ki from each record of the processing target file stored in the first area of the main memory 9, and adds the start location data (on the main memory 9) of a record

having the key field to the key field Ki as an identifier Ai. In this embodiment, since each record comprises a plurality of key fields, and each record length and each key field length are constant, the start location data (on the main memory 9) of each record can be obtained by multiplying the record length by the number of records (**Matsuda** Col 8, Lines 66-68 & Col 9, Lines 1-8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the starting location of the fields.

With respect to claim 57, **AAPA** does not explicitly teach **"the data sort method of claim 56, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record."**

However, **Matsuda** discloses **"the data sort method of claim 56, wherein the sort control card identifies a starting position for each key field in a record relative to a first key field of the record"** as the logical data of each of the processing target file and the output file include a file format, a block length, a record length, the respective key field positions and lengths of a multi-key field (**Matsuda** Col 12, Lines 10-14). Adding an identifier as a number or relative position data of each record to the key field, performing an operation designated by the operation command using an

obtained pair of the key field and the identifier as a processing unit (**Matsuda** Col 6, Lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the position of the key fields.

With respect to claim 58, **AAPA** does not explicitly teach **"the data sort method of claim 56, wherein the sort control card further indicates a data type for each key field in a record."**

However, **Matsuda** discloses **"the data sort method of claim 56, wherein the sort control card further indicates a data type for each key field in a record"** as the controller is constituted by, e.g., a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., and receives parameters from the CPU 1. The parameters include: data representing the position of a result identifier string stored in the second area of the main memory 9 (addresses in the second area of the main memory 9); logical data concerning a processing target file stored in the first area of the main memory 9 (file data such as a file format, a block length, and a record length); logical data concerning an output file (a file format, a block length, a record length, and the like); and physical data representing the storage location of an output file (output file

data such as a location on a disk and size) (**Matsuda** Col 10, Lines 52-62). Examiner interprets the file format as data type.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to rearranges the respective records in accordance with these parameters and thus forming an output file (**Matsuda** Col 10, Lines 63-68) and data type being one of the parameters.

With respect to claim 59, **AAPA** does not explicitly teach **"the data sort method of claim 58, wherein the sort control card further indicates a sort order for each key field in a record."**

However, **Matsuda** discloses **"the data sort method of claim 58, wherein the sort control card further indicates a sort order for each key field in a record"** as upon reception of an operation command for sorting (ascending order/descending order) or a relational operation from an input mechanism on a terminal side (**Matsuda** Col 9, Lines 35-37).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66).

With respect to claim 60, **AAPA** does not explicitly teach, **"the data sort method of claim 58, wherein the sort control card comprises a parameter list."**

However, **Matsuda** discloses **"the data sort method of claim 58, wherein the sort control card comprises a parameter list"** as the controller is constituted by, e.g., a 32-bit microprocessor 68020 available from MOTOROLA INC., U.S.A., and receives parameters from the CPU 1. The parameters include: data representing the position of a result identifier string stored in the second area of the main memory 9 (addresses in the second area of the main memory 9); logical data concerning a processing target file stored in the first area of the main memory 9 (file data such as a file format, a block length, and a record length); logical data concerning an output file (a file format, a block length, a record length, and the like); and physical data representing the storage location of an output file (output file data such as a location on a disk and size) (**Matsuda** Col 10, Lines 52-62).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to rearrange the respective records in accordance with these parameters and thus forming an output file (**Matsuda** Col 10, Lines 63-68).

(10) Response to Argument

A. § 103(a) rejection of claims 1-12, 16-24, 27-37, and 40-52 over AAPA in view of Matsuda.

Appellant argued that Matsuda does not teach: **"the use of variable length key information"** as required by claims 1, 16, 27 and 40.

First, the examiner would like to point out that this limitation is not being claimed in independent claims 1, 27, and 40. It is being only claimed in independent claim 16 as "plurality of key fields at least one of which is a variable length key field."

In response to the Appellant's arguments, the arguments were fully considered but were not deemed persuasive.

Examiner respectfully submits that the examiner has used AAPA to teach the argued limitation instead of Matsuda et al. AAPA teaches **"the use of variable length key information"** as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003).

Further Appellant argues that claims 1, 27 and 40 recite "expanding extracted key information," therefore these claims inherently recite data records whose key information is variable length.

In response to the preceding argument, Examiner respectfully submits that the final action taught **"extracting key information and expanding the extracted key information"** as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003).

Further Appellant argues that these references do not suggest their combination.

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case AAPA teaches the prior art techniques for sorting object records based on records sort keys.

Matsuda's invention directed towards database processing teaches the operations of sorting and the use of keys fields and identifiers to sort the records.

Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to reduce the load factors of the CPU

and the bus system and the processing performance is greatly improved by not transferring the result from main memory to the magnetic disk/intermediate storage (**Matsuda** Col 7, Lines 36-40).

Appellant's arguments directed towards the rejections of claim 2-12, 17-24, 28-37, and 41-52 reiterate deficiencies Appellant made in the rejection of the independent claims 1, 16, 27, and 40 and do not address any new points. Therefore examiner submits that if the rejection of the independent claims is deemed proper, the rejection of claims 2-12, 17-24, 28-37, and 41-52 should also be upheld.

B. § 103(a) rejection of claims 13-15, 25-26, 38-39, and 53-63 over AAPA in view of Matsuda further in view of Ferguson.

Appellant argued that Ferguson does not teach describe or suggest **"data objects that have variable length keys"** as required by claim 55.

Again the examiner would like to point out that this limitation is not being claimed in independent claim 55.

In response to the preceding argument examiner respectfully submits that the examiner has used AAPA to teach the argued limitation instead of Ferguson et al.

AAPA teaches **"the use of variable length key information"** as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003).

Examiner respectfully submits that the final action taught **"extracting key information and expanding the extracted key information"** as the act of padding converts variable length key fields to fixed length key fields of a size great enough to accommodate any value that the key may assume (**AAPA** Paragraph 0003).

Further Appellant argues that examiner has failed to state a prima facie case of obviousness.

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case AAPA teaches the prior art techniques for sorting object records based on records sort keys.

Matsuda's invention directed towards database processing teaches the operations of sorting and the use of keys fields and identifiers to sort the records.

Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Matsuda's** teaching would have allowed **AAPA** to reduce the load factors of the CPU and the bus system and the processing performance is greatly improved by not transferring the result from main memory to the magnetic disk/intermediate storage (**Matsuda** Col 7, Lines 36-40).

Ferguson's invention is directed towards sorting of data records and sorting method uses buffer size substrings to sort strings of key records.

Therefore it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Ferguson's** teaching would have allowed **AAPA and Matsuda** to require fewer storage system access and hence is generally faster (**Ferguson** Col 5, Lines 42-45) by merging the reorganized data records and to sort key records first, and then build a tree based on keys extracted at intervals from sorted key records (**Ferguson** Col 3, Lines 2-4). To efficiently execute sophisticated data processing for complex operations such as a sorting operation or a relational operation in an RDB with a simple control operation (**Matsuda** Col 1, Lines 63-66) by determining the starting location of the fields.

Appellant's arguments directed towards the rejections of claim 13-15, 25-26, 38-39, 53-54, and 56-63 reiterate deficiencies Appellant made in the rejection of the independent claim 55 and do not address any new points. Therefore examiner submits

that if the rejection of the independent claims is deemed proper, the rejection of claims 13-15, 25-26, 38-39, 53-54, and 56-63 should also be upheld.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Usmaan Saeed

Examiner

Conferees:

John Breene ,

Supervisory Patent Examiner

HOSAIN ALAM
SUPERVISORY PATENT EXAMINER

Tim Vo

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Supervisory Patent Examiner